J. Symbolic Computation (1990), **10**, 225–252

Non-Convex Contour Reconstruction

P. ALEVIZOS,*[1] J. D. BOISSONNAT†[2] AND M. YVINEC[3]

[1] *Department of Mathematics, University of Patras,* 26110 *Patras, Greece*
[2] *INRIA Sophia Antipolis,* 2004 *Route des Lucioles,* 06565 *Valbonne, France*
[3] *Laboratoire d'Informatique, Ecole Normale Supérieure,*‡ 45 *Rue d'Ulm,* 75230 *Paris, France*

(*Received* 30 *April* 1988)

We present algorithms to reconstruct the planar cross-section of a simply connected object from data points measured by rays. The rays are semi-infinite curves representing, for example, the laser beam or the articulated arms of a robot moving around the object. This paper shows that the information provided by the rays is crucial (though generally neglected) when solving 2-dimensional reconstruction problems. The main property of the rays is that they induce a total order on the measured points. This order is shown to be computable in optimal time $O(n \log n)$. The algorithm is fully dynamic and allows the insertion or the deletion of a point in $O(\log n)$ time.

From this order a polygonal approximation of the object can be deduced in a straightforward manner. However, if insufficient data are available or if the points belong to several connected objects, this polygonal approximation may not be a simple polygon or may intersect the rays. This can be checked in $O(n \log n)$ time.

The order induced by the rays can also be used to find a strategy for discovering the exact shape of a simple (but not necessarily convex) polygon by means of a minimal number of probes. When each probe outcome consists of a contact point, a ray measuring that point and the normal to the object at the point, we have shown that $3n-3$ probes are necessary and sufficient if the object has n non-colinear edges. Each probe can be determined in $O(\log n)$ time yielding an $O(n \log n)$-time $O(n)$-space algorithm. When each probe outcome consists of a contact point and a ray measuring that point but not the normal, the same strategy can still be applied. Under a mild condition, $8n-4$ probes are sufficient to discover a shape that is almost surely the actual shape of the object.

1. Introduction

Let us consider a robot equipped with a sensing device, moving around an unknown object. By means of its sensor, the robot probes the object and the problem is to reconstruct, from the probe responses, the shape or some aspect of the unknown object. A variety of subproblems can be distinguished, depending on the model of the sensor and on the constraints on the type of the object. In this paper, we restrict our attention to 2-dimensional variants of the problem and assume that the sensor probes are in a plane.

Let us consider the typical situation where each probe response consists of the coordinates of a point on the boundary of the object (a "finger probe", according to Skiena's (1988) taxonomy). In order to reconstruct the shape of the object, it is crucial to recover, from the probe responses, the order of the measured points as they appear on the boundary of the object. Clearly, without this order, it is impossible to infer the shape or even a

* This work was done while this author was visiting INRIA Sophia-Antipolis.
† This author has been partly supported by the CEE ESPRIT Project P-940.
‡ Unité de recherche associée au CNRS (URA 1327).

0747-7171/90/090225+28 $03.00/0

reasonable approximation of the shape and we must content ourselves with heuristics or with methods that are only guaranteed to work for sufficiently dense sets of points (Boissonnat, 1984; O'Rourke *et al.*, 1987). On the other hand, having this order will allow us to compute a polygonal approximation of the object by connecting the measured points in their order.

The order the points appear on the boundary of the object is, for some classes of objects, implicitly contained in the data. If the object is convex, the order is simply the order the points appear on their convex hull. The order is also implicitly contained in the data points if, as shown by O'Rourke (1986), the object is an orthogonal polygon† (i.e., each vertex is incident to exactly one horizontal edge and one vertical edge).

In order to study more general objects, we prefer not to impose constraints on the shape of the object but, instead, to use more powerful probes. This is not to say that new sensing devices are necessarily required. In fact, this paper is motivated by the observation that, in most situations, the information necessary to recover the order is implicitly contained in the probe. More precisely, the information is contained in the rays that were used to measure the points. A ray is any semi-infinite curve that has the measured point as its origin and that does not intersect the interior of the object (see Figure 1). In case of an optical device, the rays are half straight lines between the sensor (supposed to be at infinity) and the points—the optical rays. If the object is not transparent, these rays cannot intersect the interior of the object. In other situations, the robot may simply touch the object with the tip of its arm(s). In that case, the rays consist in a set of polygonal lines issued from the contact points, that represent the different positions of the arm(s) when touching the object. Notice that, in this case, the number of line segments of each polygonal ray is bounded by the maximal number of articulated bodies of the robot arm(s). A similar situation occurs when a mobile robot moves along some path until it encounters the boundary of an object. The path followed by the robot is a ray.

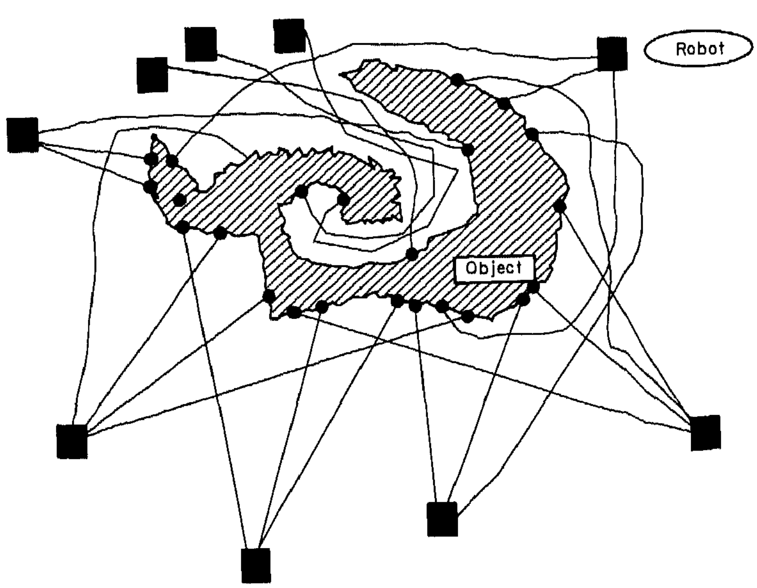

Figure 1. A robot probing an object with rays.

† In fact, O'Rourke proved the result for a family of disjoint orthogonal polygons.

The information provided by the rays is crucial for reconstructing shapes. Indeed, we will show in section 2.2 that the rays induce a total order on the points, the same as the order the points appear along the boundary of the object. Moreover, this order can be computed in optimal $O(n \log n)$ time if the number of measures is n. The algorithm presented in section 2.3 is fully dynamic and allows the insertion or even the deletion of a measured point in $O(\log n)$ time. From this order a polygonal approximation of the object can be deduced in a straightforward manner as mentioned above. However, if insufficient data are available or if the points belong to several connected objects, this polygonal approximation may not be a simple polygon or may intersect the rays. This can be checked in $O(n \log n)$ time as is shown in section 2.4.

When the object is known to be piece-wise linear, the order induced by the rays can also be used to find a good strategy for discovering the *exact* shape of the object. This problem is often referred to as the *probing problem* in the literature and has been studied in the case of a convex polygonal object: Cole & Yap (1987) showed that the shape of a convex polygon with n edges can be determined with no more than $3n$ probes; Bernstein (1986) has improved this result if the polygon is restricted to a finite set, and Dobkin *et al.* (1986) have considered the case of convex polytopes in multidimensional space. A work of synthesis of the field of geometric probing as well as a collection of new results can be found in Skiena's Ph.D. Thesis (1988). In section 3, we show how the order induced by the rays can be used to generalize the results of Cole & Yap to non-convex polygons with no colinear edges. It is proved that, if each probe outcome consists of a contact point, a ray measuring that point and the normal to the object at the point, $3n-3$ probes are necessary (section 3.4) and sufficient (section 3.2). In section 3.3, we prove that each probe can be determined in $O(\log n)$ time yielding an $O(n \log n)$-time $O(n)$-space algorithm. If the normals are not available, it is proved in section 3.5 that the same strategy can still be applied. Under a mild condition, $8n-4$ probes are sufficient to discover a shape that is almost surely the actual shape of the object.

Let us give a few preliminary notations, remarks and general assumptions that will be made throughout the whole paper. In the two dimensional plane, let P be a set of n points $p_1 \dots p_n$ and L be a set of n semi-infinite curves, called *rays*, $l_1 \dots l_n$, such that l_i originates at point p_i. According to physical constraints, we always consider that the points of P lie on the boundary of a real solid object and that the rays never intersect the interior of the object.† We assume that the n points belong to a unique simply connected object without holes. From a theoretical point of view, the ray can be any simple semi-infinite continuous curve originating at a point of P and ending at a point at infinity. The restriction to simple curves is in fact not essential and done for simplicity. For the purpose of practical reconstruction algorithms, we shall restrict our presentation to polygonal rays although the method works for more general curves; the last edges of these polygonal lines are supposed to be semi-infinite straight lines. Moreover, our complexity results assume that the number of segments of each ray is bounded by a constant.

2. The Contour Reconstruction Problem

2.1. Statement of the Problem

For the given set of points P and the set of rays L, the aim is to find a polygonal approximation of the object boundary, called a *polygonal contour* in the sequel, that is a

† For short, we will omit "*the interior of*" in the sequel.

simple polygon having the points of P as vertices and intersecting none of the rays of the set L. Such a polygonal contour does not exist for any given set of data (P, L). There are two typical situations in which no polygonal solution to the contour problem can be found.

The first one is shown in Figure 2. In this case, there are two rays, say ray l_a measuring point a and ray l_b measuring point b, that intersect in at least one point at finite distance. Such a pair of rays partition the plane into more than one connected region. Let W_{ab} be the union of the regions that do not contain points a and b; W_{ab} is the empty region if rays l_a and l_b do not intersect at a finite point and a simple wedge if l_a and l_b are straight intersecting rays. If some points, like point c in Figure 2, are measured in the region W_{ab}, then obviously the data points belong to more than one object because no continuous curve can join points in W_{ab} to points a and b without intersecting ray l_a or ray l_b. We shall say that the region W_{ab} is *hidden* by rays l_a and l_b.

In the sequel, we call *legal* a set of data (P, L) for which no point lies in a hidden region. In other words, a set of data (P, L) is legal if any pair of measured points (a, b) can be joined by a continuous curve without intersection with the set of rays L (except at points a and b with respectively the rays l_a and l_b measuring these points).

In the second situation, shown in Figure 3, the data are legal and there is a simple contour passing through all the points of P and intersecting no ray of L but this contour cannot be drawn with straight line segments joining the points of P. Such a situation arises when too few data are available: for example, in Figure 4, the addition of a new point b to the set of data restores the existence of a solution to the contour problem. In this case, we say that the contour problem admits only a *topological* solution. A topological contour on a set of points P is a cyclic ordering of the points of P such that there is a simple closed curve passing through all the points of P in that order. Such a curve is called a representation of the topological contour and is oriented counterclockwise. A topological contour becomes a linear ordering of the points as soon as one particular

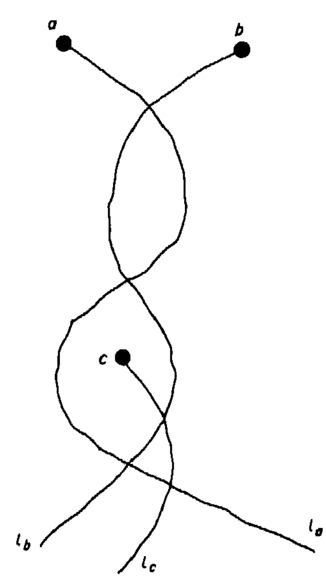

Figure 2. The case where the data points belong to more than one object.

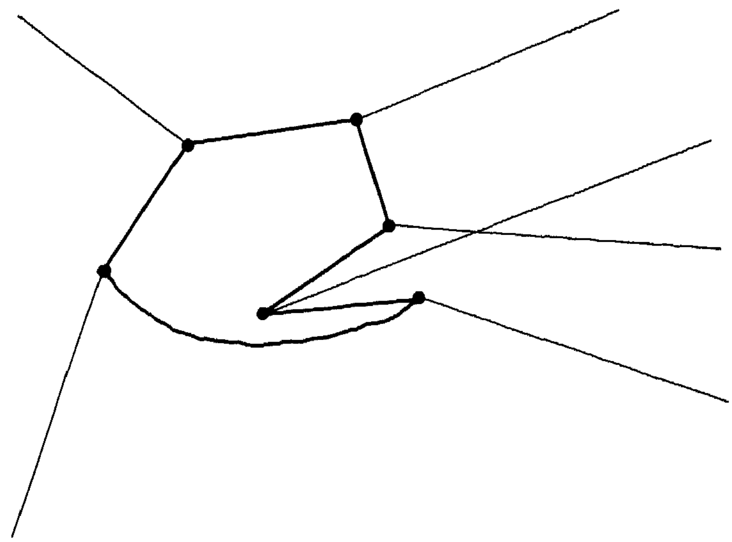

Figure 3. Too few data are available.

point has been chosen as the origin on the contour. A topological contour on a set of points P is a solution of the contour problem (P, L) if there is a representation of this contour that has no intersection with the set of rays L.

In this section, we shall first prove that any legal contour problem admits a unique topological solution. Then, we present an algorithm that finds the topological solution of a legal contour problem of size n in time $O(n \log n)$ which is shown to be optimal. Furthermore, we say that a legal set of data (P, L) is *complete* if the closed polygon that

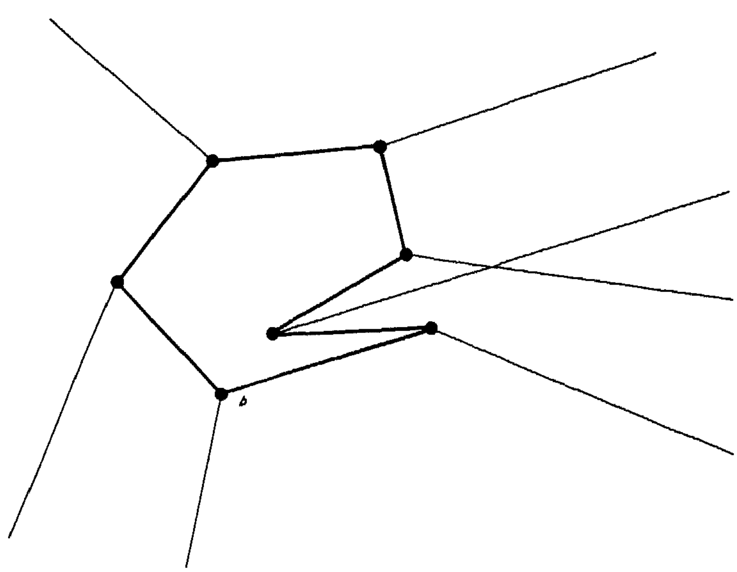

Figure 4. Adding a new point to restore the existence of a solution to the contour problem.

is the (unique) piece-wise linear representation of the topological solution is actually a solution of the contour problem. In case of illegal or incomplete data sets, this algorithm yields a polygon that either is not simple or intersects some of the rays. Both situations can be detected a posteriori through a series of simple tests performed on the obtained polygon in time $O(n \log n)$. Unfortunately, in case of failure, these simple tests do not distinguish between illegal and incomplete data.

2.2. EXISTENCE AND UNIQUENESS OF THE TOPOLOGICAL CONTOUR OF ANY LEGAL SET OF DATA

Let (P, L) be a legal set of data. We first prove a lemma that is a necessary and sufficient condition for two points a and b to be consecutive along the topological contour solution of the problem (P, L). A few definitions are needed. For any pair of points (a, b), let $\sum_{ab}$ be a simple curve joining a to b without intersecting the rays of L. The curve $\sum_{ab}$, together with the rays l_a and l_b measuring respectively the points a and b partitions the plane into three regions: the first one is the (eventually empty) region W_{ab} that is hidden by rays l_a and l_b, the other two, called H_{ab} and $\bar{H}_{ab}$, arise when splitting the complementary region of W_{ab}; H_{ab}(respectively, $\bar{H}_{ab}$) is the region to the right (respectively, to the left) of $\sum_{ab}$ assumed to be oriented from a to b (see Figure 5).

LEMMA 2.1. *Two points a and b are consecutive on any topological contour solution of the problem (P, L) if and only if there exists a simple curve $\sum_{ab}$ such that:*

(i) *the curve $\sum_{ab}$ intersects no ray of L;*
(ii) *the region H_{ab} contains no point of P in its interior.*

PROOF. Assume first that a and b are consecutive points on a topological contour solution of the problem (P, L). Let $\sum$ be a representation of this topological solution and let $\sum_{ab}$ be the part of $\sum$ joining a to b. Then, by definition, $\sum_{ab}$ intersects no ray of L and furthermore the interior of the region H_{ab} is totally included in the outside of the object bounded by $\sum$ and thus contains no point of P.

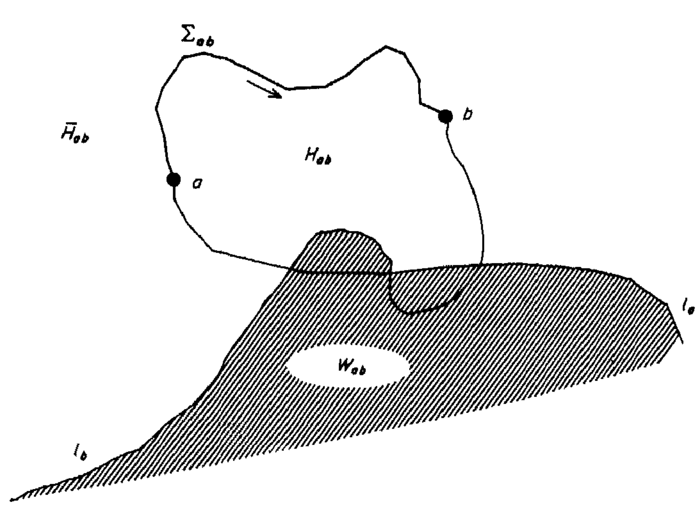

Figure 5. For the definition of W_{ab}, H_{ab} and $\sum_{ab}$.

Conversely, assume that there is a curve $\sum_{ab}$ joining a to b and satisfying both conditions of Lemma 2.1. Assume for a contradiction that there is a solution of the problem (P, L) in which a and b are not consecutive. let $\sum'$ be a representation of this topological solution; then $\sum'$ goes at least through another point c of P between a and b; let $\sum'_{acb}$ be the part of $\sum'$ joining a to b (see Figure 6). The point c, which cannot belong to the hidden region W_{ab}, either belongs to H_{ab}, which contradicts the second condition of Lemma 2.1, or belongs to $\bar{H}_{ab}$. In this latter case, the ray l_c, measuring c, that is on the right side when going from a to b on $\sum'_{acb}$, must necessarily intersect $\sum_{ab}$ which contradicts the first condition of Lemma 2.1.

A direct consequence of this lemma is the following corollary:

COROLLARY 2.2. *When a topological solution for the contour problem (P, L) exists, it is unique.*

Lemma 2.1 can be generalized in a straightforward way. Let $\sum_{q_1 \ldots q_m}$ be a simple curve joining the points $q_1, \ldots, q_m$ of P in that order. A necessary and sufficient condition for $\sum_{q_1 \ldots q_m}$ to be part of a representation of the topological solution of the problem (P, L) is that:

(i) $\sum_{q_1 \ldots q_m}$ intersects no ray of L (except the ray l_i measuring point q_i at this point for $i = 1, \ldots, m$);

(ii) $H_{q_1 \ldots q_m} = \bigcup_{i=1}^{m-1} H_{q_i q_{i+1}}$ contains no point of P.

Let us come now to the existence of a topological solution for any legal contour problem. More precisely, we shall prove the following theorem:

THEOREM 2.3. *For any legal set of data (P, L), the set of rays L induces a total cyclic ordering of the points of P. This total cyclic order is the topological solution of the contour problem (P, L).*

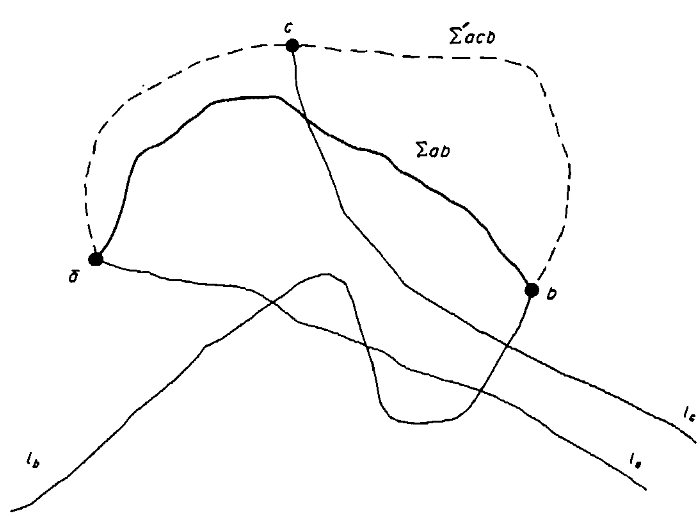

Figure 6. For the proof of Lemma 2.1.

PROOF. Let us choose the point a of P as the origin. For any point q_i of $P-\{a\}$, let $\sum_{aq_i}$ be a simple curve joining a to q_i without intersecting the rays of L. For any pair (q_i, q_j) of points of $P-\{a\}$, either q_i lies inside the region H_{aq_j} or q_j lies inside the region H_{aq_i}. Indeed, assume that q_j does not lie inside the region H_{aq_i}, then q_j lies in $\bar{H}_{aq_i}$ because the set of data is legal. Now the ray l_j of q_j does not intersect $\sum_{aq_i}$ which implies that q_i lies in the region H_{aq_j} (see Figure 7). We shall denote by $q_i <_a q_j$ the relation "*q_i lies in H_{aq_j}*". Because the data are legal, no point of P lie in the hidden region $W_{q_iq_j}$. Furthermore, if the two curves $\sum_{aq_i}$ and $\sum_{aq_j}$ intersect each other at some points distinct from a, no point of P can lie in the interior of a region totally bounded by these two curves because the ray of such a point would necessarily intersect $\sum_{aq_i}$ or $\sum_{aq_j}$. Thus, the relation $q_i <_a q_j$ is equivalent to the fact that the subset $P \cap H_{aq_i}$ is strictly included in the subset $P \cap H_{aq_j}$ which proves the transitivity of this relation. The relation $<_a$ is therefore a total order relation on the points of $P-\{a\}$.

The order relation $<_a$ is independent of the actual choice of the curves $\sum_{aq_i}$ and $\sum_{aq_j}$ provided that they do not intersect the rays of L. Indeed any choice of these curves yields the same subsets of points $P \cap H_{aq_i}$ and $P \cap H_{aq_j}$ because no point of P lie in the interior of a region of the plane totally bounded by these curves. Let us now show that this ordering is the topological solution of the contour problem. Let q_1 be the minimum of $P-\{a\}$ for the order $<_a$; then obviously H_{aq_1} contains no point of P and q_1 is the unique point of P having this property. Then, from Lemma 2.1, we know that the successor of a on the contour is q_1. Now for any pair (q_i, q_j) of points of $P-\{a\}$ such that $q_i <_a q_j$ and for any curve $\sum_{q_iq_j}$ not intersecting the rays, we have (see Figure 8):

$$P \cap H_{aq_j} = (P \cap H_{aq_i}) \cup \{q_i\} \cup (P \cap H_{q_iq_j}),$$

where the unions are disjoint unions. This is easily shown using the same kind of arguments as above. Equivalently, if *card*( ) is the number of elements of a set, we have:

$$card(P \cap H_{aq_j}) = card(P \cap H_{aq_i}) + 1 + card(P \cap H_{q_iq_j}).$$

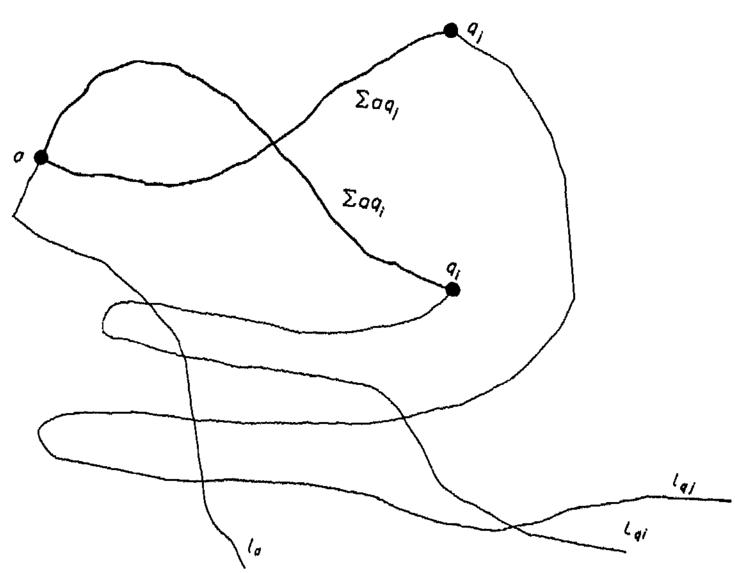

Figure 7. q_i lies in the region H_{aq_j}.

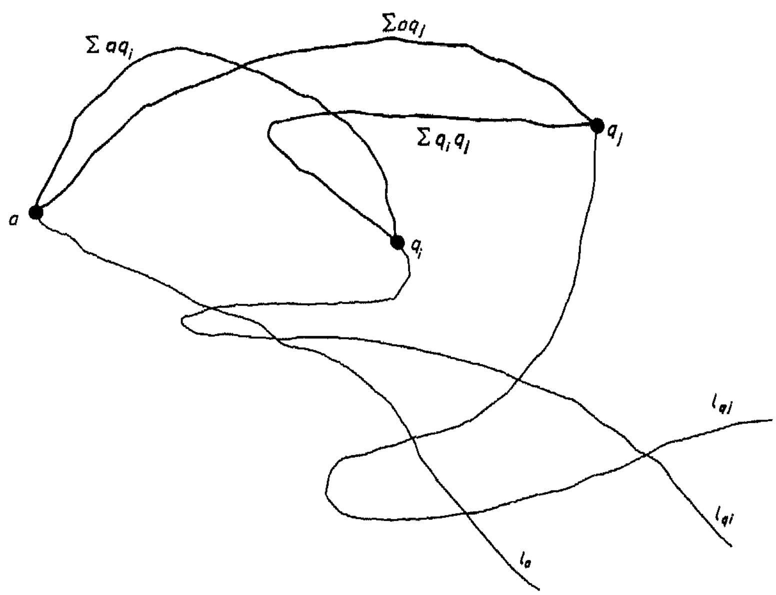

Figure 8. The order relation is independent of the actual choice of Σ_{aq_i} and Σ_{aq_j}.

Then the successor of q_i in the order $<_a$ is the unique point q_j such that

$$card(P \cap H_{aq_j}) = card(P \cap H_{aq_i}) + 1,$$

which is equivalent to say that $card(P \cap H_{q_iq_j}) = 0$. Thus, from Lemma 2.1, the successor of q_i on the topological solution of the problem (P, L) is uniquely defined as being the successor of q_i in the order $<_a$.

2.3. AN $O(n \log n)$ ALGORITHM FOR THE CONTOUR RECONSTRUCTION PROBLEM

In this section we propose an algorithm that, for any legal set of data (P, L) provides the ordering of the points of P corresponding to the topological solution of the contour problem (P, L). This algorithm, called *Algorithm Contour*, is in fact an incremental sorting algorithm, where the data points are introduced one by one into a balanced tree structure (e.g. an AVL tree), that maintains the points processed so far, in the order induced by their measuring rays. The data structure uses $O(n)$ space and the whole algorithm runs in time $O(n \log n)$ which is shown to be optimal. Moreover, the data structure is fully dynamic and allows the insertion as well as the deletion of data points in time $O(\log n)$ per insertion or deletion. Our analysis assumes that the rays are polygonal lines with a bounded number of edges, the last edge of each ray being a semi-infinite straight line. In fact, the same algorithm and its complexity analysis can be applied to more general ray curves as long as two rays intersect in at most a bounded number of points each of which can be found in constant time.

The correctness of this incremental algorithm relies on the fact, stated in Lemma 2.4 below, that the topological solution of the contour problem for any subset (P', L') of the legal set of data (P, L) is a subsequence of the topological solution of the problem (P, L).

LEMMA 2.4. *Let (P', L') be any subset of the legal set of data (P, L). The topological solution of the contour problem (P', L') is a subsequence of the solution of the problem (P, L).*

PROOF. Let a be a point of P chosen as the origin. Let q_i and q_j be two points of P' such that q_i precedes q_j in the order induced on the set of points P by the set of rays L ($q_i <_a q_j$). Then for any choice of the curves $\sum_{aq_i}$ and $\sum_{aq_j}$ that do not intersect the rays of the set L, we have:

$$P \cap H_{aq_i} \subset P \cap H_{aq_j},$$

which implies that

$$P' \cap H_{aq_i} \subset P' \cap H_{aq_j}.$$

This, from the proof of Theorem 2.3, is equivalent to saying that q_i precedes q_j in the order induced by the subset of rays L' on the subset of points P'.

Let us now present one of the basic ingredients of our method. It is an elementary algorithm, called Function Threepoints, which solves the contour problem for a reduced data set consisting of three measured points.

Let a, b, c be three points measured, respectively, by the rays l_a, l_b and l_c that together form a legal data set. Function Threepoints will answer the following question: which one of the two sequences *abc* or *acb* is the solution of the contour problem $(\{a, b, c\}, \{l_a, l_b, l_c\})$. To answer this question, *Function Threepoints* searches which one of the two points b or c is the successor of point a on the contour by constructing a simple curve $\sum$ that joins point a to either b or c and satisfies the condition of Lemma 2.1.

The curve $\sum$ is constructed as follows: let us consider the generalized arrangement formed by the three rays l_a, l_b and l_c together with the line at infinity Γ. Γ is a closed curve that encloses all the points of P and all the intersections between the rays. This arrangement is made of generalized edges that are the connected portions delimited on each of these four curves by its intersections with the three others. The curve $\sum$ follows some of the edges of this arrangement always staying at a small distance ε of the rays and turning left each time an intersection point is encountered. More precisely, the curve $\sum$, starting from point a, follows ray l_a towards infinity at distance ε on the left side of l_a (assumed to be oriented from a to infinity) until it reaches (at distance ε) the first intersection point on l_a. At this point, $\sum$ turns left and then follows the intersecting curve at distance ε until the next intersection point is reached and so on. (See Figure 9 for examples.)

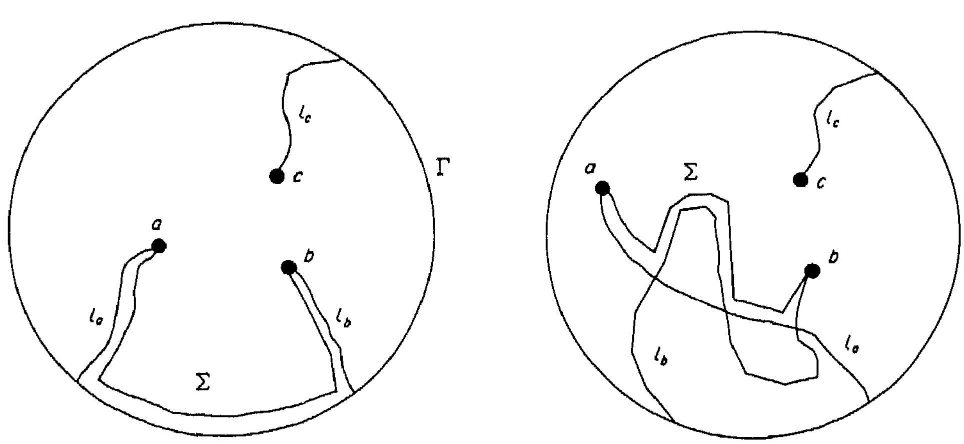

Figure 9. Illustrations of the construction of $\sum$ by Function Threepoints.

By construction, the curve $\sum$ is a simple curve that follows each edge of the arrangement at most once and thus cannot return to the point a because a is the origin of a ray. As the number of edges of each ray is bounded by a constant, the same holds for the number of edges in the arrangement. Thus $\sum$ will necessarily reach either point b or point c after a finite bounded number of left turns.

The curve $\sum$ intersects none of the rays l_a, l_b and l_c and delimit with the rays measuring its endpoints a region H that is such that $H-(W_{ab} \cup W_{bc} \cup W_{ca})$ is arbitrarily small and thus includes no point of the three points set. Thus, it follows from Lemma 2.1, that the point reached by $\sum$ is the successor of point a on the contour.

From the complexity point of view, since the number of edges of each ray is bounded, the number of intersections between two rays is also bounded and the above arrangement, together with the path $\sum$, can be constructed in constant time. This discussion can be summarized by the following pseudo-code for Function Threepoints and the subsequent Lemma:

FUNCTION THREEPOINTS $(\{a, b, c\}, \{l_a, l_b, l_c\})$
input: *three points a, b, c and their measuring rays l_a, l_b, l_c forming together a legal data set.*
output: *the solution of this three point contour problem.*

(i) Construct the arrangement formed by the three rays l_a, l_b, l_c together with the line at infinity Γ;

(ii) construct the path $\sum$ joining a to either b or c as described above;

(iii) if $\sum$ reaches point b the solution is abc else the solution is acb.

LEMMA 2.5. *Function Threepoints provides the topological solution of a three points contour problem in constant time if the number of intersections between two rays is bounded and if each one of these intersections can be found in constant time.*

Let us come now to the description of *Algorithm Contour.* In this algorithm, the Function Threepoints plays the role of the comparison function that is needed by any sorting algorithm. Indeed, in view of Lemma 2.5, a call to this function for any three points subset $(\{a, b, c\}, \{l_a, l_b, l_c\})$ of (P, L) provides the ordering of the three points a, b and c on the contour solution of the problem (P, L): in other words, this function compares b to c according to the total order induced by the set of rays L once point a has been chosen as its origin.

As previously noticed, Algorithm Contour is an incremental sorting algorithm that processes in turn each data point. The first processed point, say a, is chosen as the origin of the contour.

The current set of data points (except a) is maintained as a balanced binary tree, e.g. an AVL tree, called the *contour tree.* Each node in this tree stores one of the already processed data points and a symmetrical traversal of the contour tree provides the ordering of the set of data points along the contour, taking a as the origin.

When inserting a new data point x, we traverse the tree from the root to a leaf. At each visited node, say node V storing the data point v, Function Threepoints is called with the $(\{a, x, v\}, \{l_a, l_x, l_v\})$ as its argument. If the function returns the sequence axv, we visit the left son of node V, otherwise, we visit the right son of V. When a leaf is reached, a new node is created to store x either as a left or a right son, depending on the ordering returned by the last call to Function Threepoints. The tree is then rebalanced.

When all the data points have been inserted, a symmetrical traversal of the tree gives the solution of the contour problem.

From the above discussion, it is clear that Algorithm Contour provides the topological solution of the contour problem (P, L) if the data are legal. The analysis of the worst case complexity of this algorithm is straightforward and yields the following theorem:

THEOREM 2.6. *Algorithm Contour provides the topological contour solution of a legal set of data* (P, L) *of size* n *in time* $O(n \log n)$. *The data structure uses* $O(n)$ *space and may be updated in time* $O(\log n)$ *when adding or deleting a point from the data set. These results are asymptotically optimal in the worst-case.*

As the insertion, the deletion of a node from the contour tree is performed using standard techniques on balanced trees. When the origin is to be deleted from the data set, we take its successor as the new origin and delete the corresponding node from the contour tree.

The optimality of the algorithm comes from the fact that sorting is reducible in $O(n)$ time to the contour problem. Consider a set of n points $p_1, \ldots, p_n$ pictured in Figure 10, that consists of $n-1$ collinear points and another, say q, not on the same line. Consider the corresponding rays $l_1, \ldots, l_n, l_q$, taken to be parallel half straight lines, normal to the line $p_1, \ldots, p_n$; $l_1, \ldots, l_n$ lie in the half plane not containing q; the direction of l_q is opposite to the direction of $l_1, \ldots, l_n$. The edge list produced by an algorithm solving the contour problem can be used to sort the p_i in $O(n)$ additional operations.

2.4. TESTING A POSTERIORI THE SOLUTION OF THE CONTOUR PROBLEM

If the set of data (P, L) is not legal there exists no solution to the contour problem, not even a topological solution. If the data are legal but not complete, the solution of the contour problem is only topological and cannot be represented by a simple polygon. In both cases, the ordering obtained by a blind application of the above algorithm corresponds to a polygon that either is not simple or intersects some of the rays. The simplicity of the contour can be tested in $O(n \log n)$ time using a *plane sweep* algorithm to detect an eventual intersection between the edges, such as the *line segment intersection test* described by Shamos & Hoey (1975).

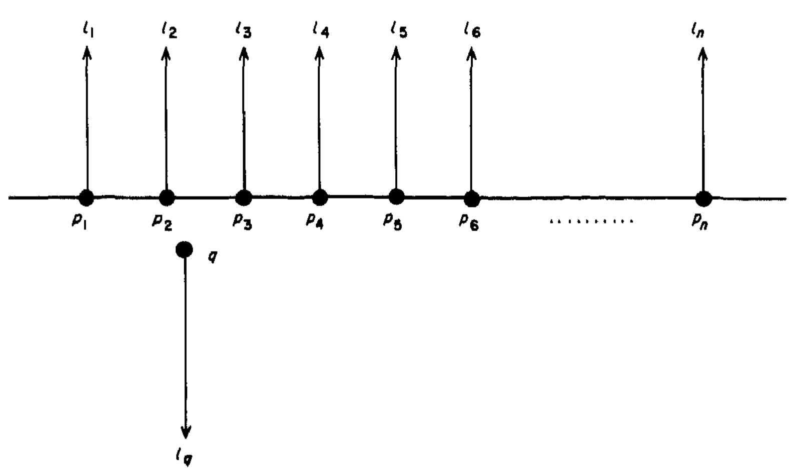

Figure 10. For the lower bound.

In order to check if a ray intersects the contour, we use the following result of Chazelle & Guibas (1985). There exists an $O(n)$ space data structure representing a simple polygon Π that can be computed in time $O(n \log n)$ and that, given a pair (p, u) of a point p and a direction u, can be used to find the first edge of Π hit by the ray from p in the direction u in time $O(\log n)$.

We apply this result to all the segments of all the rays of L. Let s be such a segment, belonging to some ray l_p, issued from point p; let a be the end-point of s encountered first when the ray is traversed from p, and b the other end-point of s. We apply the result of Chazelle & Guibas to the pair $(a, a \rightarrow b)$, where $a \rightarrow b$ is the direction of the half-line δ issued from a and containing b, and find in $O(\log n)$ time the first edge, say e, of the contour hit by δ. If the intersection between e and δ belongs to s, we have found a ray that intersects the contour; the solution is not valid. Otherwise, we consider another ray segment. Because the number of segments per ray is bounded by a constant, this test takes at most $O(n \log n)$ time. We have shown:

THEOREM 2.7. *The validity of the solution to a contour problem can be checked in* $O(n \log n)$ *time.*

Unfortunately, in case of failure of one of the above tests (test for simplicity and test for non-intersection with the set of rays), it is not possible in time $O(n \log n)$ to make the distinction between the case of illegal data and the case of incomplete data. This point is discussed at length, in a companion paper, for the case of straight line rays (Alevizos *et al.*, 1988).

REMARK. An important practical case of legal but incomplete data arises when the robot has only seen or touched too small a portion of the object (see Figure 11). In that case,

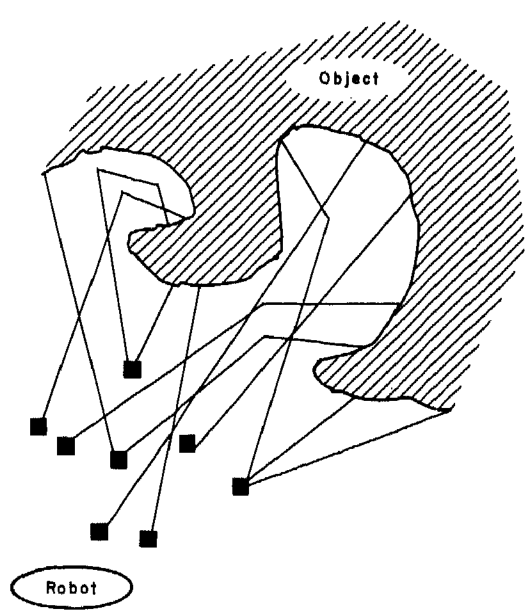

Figure 11. An important case of legal but incomplete data.

the topological solution of the contour problem can be represented by an open polygonal line including all the measured points and closed by an additional curve. In a previous paper (Alevizos *et al.*, 1987) we have shown in detail how to slightly modify Algorithm Contour to handle this case.

3. Probing Simple Polygons

3.1. Statement of the problem

Section 2 was concerned with finding a polygonal approximation of an arbitrary object shape from a set of given probes. Though the algorithm of section 2 was dynamic and allowed insertions and deletions of points, the probes were imposed and the algorithm did not take care of determining the probes. In this section, we want to discover the exact shape of an object, known to be a simple but not necessarily convex polygon, by means of a minimum (finite) number of on-line probes. "On-line" here means that the sequence of probes is determined by the algorithm, and that each individual probe is computed from the knowledge provided by the previous probes.

Our probe model is the following. One probes along a half line, called the *probe path*, whose origin is some point O_i of the plane. When the probe is issued, the probing device responds with the first point p_i, called the *contact point*, where the probe path encounters the boundary C of the object and gives also the normal n_i to C at p_i when it is defined. The sensory device is supposed to be able to detect when p_i is a vertex of C, in which case the object responds with two normals instead of one, namely the normals to the edges incident to p_i. An example of such a device may be a finger with a tactile sensor at its tip.

In addition, we make the realistic assumption that, when the probe path contains an edge of C (we call such a probe a *tangent probe*), no contact point on this edge is reported: the device misses the edge.

In the next subsection (subsection 3.2), we show that, under some mild conditions stated below, we can define a probing strategy that ensures to fully discover the exact shape of the object in at most $3n-3$ probes where n denotes the number of edges of the object. It is important to realize that n is a priori unknown and will be discovered at the same time as the exact shape of the object. Here are the required conditions:

CONDITION 3.1. The oriented supporting lines of the edges of C are all distinct.† Notice that two supporting lines may be identical if their orientations are opposite.

CONDITION 3.2. A point t of the object, called the target point, is given.

These two conditions are made to ensure that the probing problem can be solved in a finite number of steps. Indeed, without the first condition, a small detail of the object may still have been missed after any finite number of probes. Another way to circumvent this difficulty, that we do not follow here, would be to assume that the edges of the object have at least a minimal finite length. The second condition allows one to isolate the problem of discovering the shape of the object from the problem of locating it within the workspace. Without this condition, we have no idea where C is located and an unbounded number of probes can be required to find it.

† C is supposed to be oriented counterclockwise and the edges and their supporting lines accordingly.

In the following subsections, we show that each successive probe needed in this strategy can be determined in time $O(\log n)$ (subsection 3.3) and that our strategy is optimal in the sense that $3n-3$ probes may be actually necessary in some adversary worst case (subsection 3.4).

The kind of response assumed here requires a powerful probing device; previous work on the subject (Cole & Yap, 1987; Skiena, 1988) has assumed simpler probes, most notably "finger probes", that return only the first contact point. In section 3.5, we show that, under a mild condition, our algorithm can work for those simple probes. However the result is of a probabilistic nature; more precisely, we show that, under that condition, $8n-4$ probes are sufficient to discover a shape that is almost surely the actual shape of the object.

A last point about this problem, that is not considered here, is the problem of optimizing the trajectory of the robot. Indeed, we show here that each new probe can be computed in time $O(\log n)$ but that does not presume anything about the time needed by the robot to move to the right position O_i and to execute such a probe, except if the robot can jump from one position to another in $O(1)$ time, as can do a robot moving in 3-dimensional space but constrained to probe in a plane, in which case each probe can be specified and executed in $O(\log n)$ time.

3.2. UPPER BOUND ON THE NUMBER OF PROBES

Our probing strategy is based on the use of the total order induced on the set of contact points by the set of probe paths. To make use of the results of the previous section, each new probe is chosen so that the outcoming contact point p_i can be associated with a semi-infinite ray l_i ending at p_i and known not to intersect the interior of the object. The origin O_i of the current probe path is chosen to be either a point at infinity or to belong to a previous probe path. Ray l_i is the concatenation of a prefix (made of portions of previous probe paths) and of an extra segment, called the *probe segment*, that is the portion of the current probe path connecting O_i to p_i. In the sequel, we shall consider that a probe outcome, noted $\varpi_i = (p_i, n_i, l_i)$, includes three components: the contact point p_i, the normal n_i to the boundary C of the object at p_i and the semi-infinite ray l_i ending at p_i.

Given a probe outcome $\varpi_i = (p_i, n_i, l_i)$, we call the line D_i, normal to n_i and passing through p_i, the *supporting line* of ϖ_i. When necessary, D_i will be oriented so as to let l_i on its right side in the neighbourhood of p_i. If p_i belongs to the edge e_j of C, we say that e_j has been *discovered.*

Let us consider a set of s probes whose outcomes $\varpi_1, \ldots, \varpi_s$ are indexed according to the order induced by the corresponding rays $l_1, \ldots, l_s$. If at least one probe had been performed on each edge of C, it would be easy to obtain C by the following simple procedure A, that computes a list V containing the ordered set of vertices of C (indices are taken modulo s):

PROCEDURE A

1. $V := \varnothing$;
2. **for** $i = 1, \ldots, s$ **do** if $D_i \neq D_{i+1}$ **then** add $D_i \cap D_{i+1}$ to V;
3. **end.**

However, if some edges have not been discovered yet, Procedure A yields vertices that do not belong to the object. Roughly speaking, our strategy consists in issuing probes that aim these potential vertices in order to either confirm them as actual vertices of the object or to discover new edges.

We start with three probes. The first two probes are performed along straight line rays with opposite directions and both passing through the target point. Let D_1 and D_2 denote the two supporting lines of the two corresponding probe outcome ϖ_1 and ϖ_2, $I = D_1 \cap D_2$ (possibly at infinity). The third probe is performed along a directed straight line passing through the target point and I (directed in such a way that the target point is reached before I). The three corresponding contact points p_1, p_2, p_3 belong to three distinct edges of C.

At a given stage of the algorithm some edges have been discovered. The rays associated with the probes induce an ordering of the contact points and also of the discovered edges (the ordering along C). The intersection I between the supporting lines D_1 and D_2 of two successive contact points is called a *corner* and is a potential vertex of C. The algorithm maintains an ordered list of corners L and, at each step, constructs a new probe path that will either confirm the first corner I of L as being a vertex of C, or will probe a new point on a not yet discovered edge. In the first case, we simply report the vertex and delete I from L; in the latter, two new corners are discovered and are inserted in L. The algorithm halts when L is empty.

Let $I = D_1 \cap D_2$ be the current first corner of list L. The two lines define four wedges R (with p_1 and p_2 on its boundary), S (with p_1 but not p_2 on its boundary), T (with neither p_1 nor p_2 on its boundary) and U (with p_2 but not p_1 on its boundary). Let $\varpi_1 = (p_1, l_1, n_1)$ and $\varpi_2 = (p_2, l_2, n_2)$ be the two probe outcomes whose supporting lines are D_1 and D_2 and let e_1 and e_2 be the edges of C containing p_1 and p_2 respectively. The two points p_1 and p_2 are adjacent in the order induced by the set of rays, at this stage of the algorithm. Therefore, from Lemma 2.1, the regions $H_{p_1p_2}$ (or H_{12} for short) limited by the ray l_1, the ray l_2, the portion C_{12} of the boundary of C between p_1 and p_2, and lying to the right when traversing C_{12} from p_1 to p_2, is known to contain no contact point of the previous probes and no confirmed vertex. Furthermore, the contact point p of a probe ($p \neq p_1$ and $p \neq p_2$), is to be inserted between p_1 and p_2 on the boundary of C if and only if p lies inside H_{12}.

Notice that p cannot belong to l_1 nor to l_2 since otherwise it would have been already found by a previous probe.

We denote by $h_{p_1p_2}$ (or h_{12} for short) the set of points that belong to the boundary of H_{12} (considered as a closed region) but not to C_{12}. h_{12} is considered to be a connected curve (which may include an edge at infinity) oriented from p_2 to p_1.

Our aim now is to exhibit probe paths that will either confirm I as being a vertex of C or will discover a new edge lying between p_1 and p_2 on the contour C. For that purpose, we first issue a probe path μ that satisfies the three following requirements:

(i) μ passes through I in order to decide whether this point is actually a vertex or not,

(ii) μ does not intersect the supporting lines D_1 nor D_2, to avoid useless probes with contact points on already discovered edges,

(iii) the probe segment is guaranteed to lie entirely inside H_{12}, to ensure that the contact point will lie between p_1 and p_2.

Let D be a straight line passing through I and contained in $R \cup T$. D intersects the segment p_1p_2. We orient D so that p_1 is on the left side of D and p_2 on its right side. Let $\gamma = h_{12} \cup C_{12}$. γ is a simple closed curve (possibly containing points at infinity). From the Jordan theorem, D intersects γ at an even number of points $q_1, \ldots, q_{2k}$.† In the case that h_{12} contains the point at infinity of D, one intersection point is at infinity: we take it to be q_1. Let τ_i $(i=1,\ldots,2k)$ be the vector tangent to γ at q_i, oriented in the same way as h_{12}, C_{12} and thus γ. To each intersection point q_i $(i=1,\ldots,2k)$, we associate a sign, $+$ or $-$, according as the orientation of the frame (D, τ_i) is positive or not (see Figure 12). Such a sign convention implies that the oriented line D enters H_{12} at an intersection with sign $+$ and leaves it at an intersection with sign $-$. From the Jordan theorem, the sequence of signs $\sum(\gamma)$ of the intersections between γ and D, sorted along D, is an alternating sequence of $+$ and $-$: $+-+-+-\cdots+-$.

h_{12} is a simple piece of curve joining the two points p_1 and p_2. Because p_1 and p_2 are on different sides of D, it follows from the Jordan theorem that h_{12} intersects D in an odd number of points. Let $\sum(h_{12})$ be the corresponding sequence of signs. The number of $+$ signs in $\sum(h_{12})$ is exactly one more than the number of $-$ sign. ($\sum(h_{12})$ necessarily starts with a $+$ sign because H_{12} lies on the right side of C_{12} (oriented from p_1 to p_2) and thus D intersects h_{12} before it intersects C_{12}.) Hence, $\sum(h_{12})$ either terminates by a $+$ sign (Case 1) or contains two consecutive $+$ sign (Case 2). Because the sequence of signs of γ is an alternating sequence of $+$ and $-$, there exist in both cases two intersection points, successive along D, such that the first one is an intersection point between D and h_{12} and has $+$ sign and the second one is an intersection between D and C_{12} and has $-$ sign. We rename these two points O and p respectively. Notice that, in Case 1, O is the

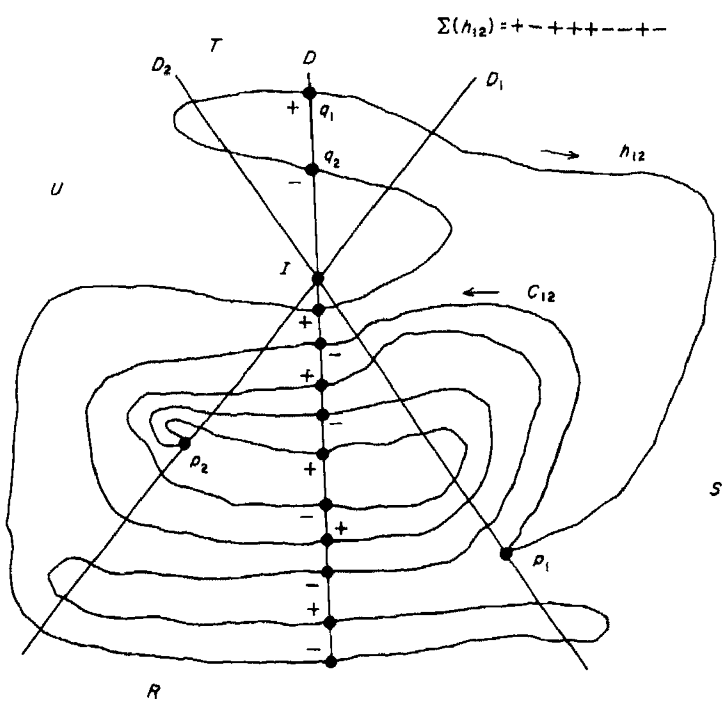

Figure 12. The sequence of signs.

† We assume that D intersects γ properly; otherwise, we slightly move D.

last intersection point between h_{12} and D; in Case 2, O is the first of two successive intersection points with signs +; thus O can be determined from h_{12} and D.

Let μ be the half line supported by D, with the same orientation as D and starting at O. We associate with μ a probing ray l that is exactly the portion of μ between O and p if O is a point at infinity and, otherwise, the concatenation of μ with the infinite portion of the ray l_i ($i = 1$ or 2) passing through O. From the above discussion, it is clear that μ intersects the boundary of C for the first time at point p and that p belongs to H_{12}. So p_1, p, p_2 are encountered in that order along the boundary of C. Let $\varpi = (p, l, n)$ be the corresponding probe outcome.

We distinguish four possible cases, depending on whether p belongs to e_1, e_2, both or none. Notice that, due to Condition 1 above, p belongs to e_i iff p belongs to D_i and $n = n_i$.

CASE 1. $p \in e_1$ and $p \in e_2$
In this case, $p = I$. I is confirmed as a vertex of C. Due to Condition 1, we are guaranteed that the edges containing p_1 and p_2 are adjacent along C and that I is their common vertex.

CASE 2. $p \notin e_1$ and $p \notin e_2$
Due to the construction of μ, $p \in H_{12}$ and thus, from Lemma 2.1, it lies on the portion of C between p_1 and p_2. Because p_1 and p_2 were consecutive in the order induced by the rays, point p belongs to an edge not yet discovered.

CASE 3. $p \in e_1$ and $p \notin e_2$
Thus $p = I$ but is not a vertex of C (this cannot occur if C is convex). In this case, we have not confirmed I as a vertex of C nor have we discovered any new edge. Another probe is needed and we exhibit a new probe path μ' that is guaranteed to discover a new edge of the boundary of C between p_1 and p_2. The new probe μ' will be supported by a straight line D' passing through I and contained in $S \cup U$.

Let Π_1 be the half-plane on the right side of D_1, when oriented as described in section 3.1. We distinguish two subcases according to whether p_2 belongs to Π_1 or not.

Subcase 3.1. $p_2 \in \Pi_1$
The situation is depicted in Figure 13. In this case, D' is oriented from S to U. Let μ' be the half line supported by D' and starting at I. The contact point probed by μ' is p'. The corresponding ray l' is the concatenation of Ip' and l. Using arguments similar to those of Case 2, the new probe necessarily discovers a new edge of C (between p_1 and p_2).

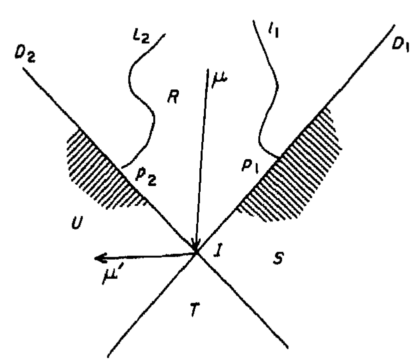

Figure 13. Case 3.1.

Subcase 3.2. $p_2 \notin \Pi_1$

The situation is depicted in Figure 14. We now orient D' from U to S. As in Case 2, among the intersections between D' and the closed Jordan curve $h_{12} \cup C_{12}$, there exist two successive intersection points along D' such that the first point, O', is an intersection point with + sign between D' and h_{12} and the second one, p', is an intersection point with − sign between D' and C_{12}. We take O' as the origin of μ'. The contact point probed by μ' is p'. The corresponding ray l' is simply $O'p'$ if O' is at infinity or the concatenation of $O'p'$ with the infinite part of the ray passing through O' otherwise. Using arguments similar to those of Case 2, the new probe necessarily discovers a new edge of C (between p_1 and p_2).

CASE 4. $p \notin e_1$ and $p \in e_2$

This case is analogous to the previous one. The indices 1 and 2 have simply to be exchanged as well as the wedges U and S.

In conclusion, each time a corner is checked, we either confirm the corner as a vertex of C by means of one probe and this corner will never be probed again or we discover a new edge by means of at most two probes. Thus to discover C we need at most one probe per vertex and two probes per edge, except for the first three edges that are discovered by means of only one probe each. If C is convex, cases 3 and 4 cannot occur and thus each probe either confirms a vertex or discovers a new edge. This proves the following theorem:

THEOREM 3.3. $3n-3$ *probes are sufficient to determine the exact shape of a simple non convex polygon C with n non-colinear edges. If C is convex $2n$ probes are sufficient.*

3.3. COMPLEXITY ANALYSIS

The above strategy guarantees that a finite number of probes are performed. However, in order to achieve an effective algorithm, we need to be precise as to how to construct the probe paths. We show now that we can restrict ourselves to polygonal rays and that each ray can be determined in $O(\log n)$ time yielding an overall $O(n \log n)$ time algorithm.

More precisely, at each step, the new probe path μ and, eventually, the additional probe path μ' are constructed as described in the previous section. We take a straight

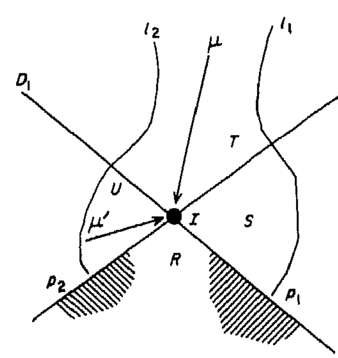

Figure 14. Case 3.2.

oriented line D contained in $R \cup T$. μ is a half line supported by D, with the same orientation as D. Its origin O is an intersection point between D and h_{12} with sign $+$ that either is the last intersection between D and h_{12} or immediately precedes another intersection between D and h_{12} with sign $+$. The additional probe path μ', if needed, is taken to be supported by a straight line D' contained in $S \cup U$. The origin O' is either the corner I (subcases 3.1 and 4.1) or a point defined in a way similar to the way O has been defined above (we simply have to exchange the roles of D and D').

LEMMA 3.4. *At each step of the algorithm, h_{12} is a polygonal convex curve (i.e. the angle between two successive segments of h_{12}, in the order they are encountered when going from p_2 to p_1, is less than* 180 *degrees).*

PROOF. The proof is by induction on the number of steps (one step corresponding to one of the cases 1, 2, 3 or 4).

Clearly, the lemma holds after the initialization. Let us suppose that k steps have been performed and that the above claim holds. Let us consider now step $k+1$.

Assume first that step $k+1$ corresponds to case 1 or 2. Once the origin O of μ has been chosen (as described in the previous section), μ is a half line with a point O of h_{12} as its origin and lying on the right side of h_{12} (oriented as usual from p_2 to p_1), in the neighbourhood of O. The new probe path μ, ending at the contact point p, splits the polygonal line h_{12} into two new polygonal lines h_{1p} and h_{p2}. The angles arising on each of those lines are exactly the angles of h_{12} except at vertex O where the angle arising on h_{1p} and the angle arising on h_{p2} sum to the angle arising on h_{12}. Thus these two new angles are also convex. The same arguments hold for both probes in subcases 3.1 and 3.2 and similarly in subcases 4.1 and 4.2. The first probe path splits the polygonal chains h_{12} into two polygonal chains h_{1I} and h_{I2}. The second probe μ' will further split either h_{I2} (subcases 3.1 and 3.2) or h_{1I} (subcases 4.1 and 4.2) into two convex polygonal chains. This completes the proof.

A direct consequence of the fact that h_{12} is a convex curve is that, either D intersects h_{12} only once or the first two intersections between D and h_{12} along D have $+$ signs. Similarly, in subcases 3.2 and 4.2, either D' intersects h_{12} only once or the first two intersections between D' and h_{12} along D' have $+$ signs. This proves the following lemma:

LEMMA 3.5. *We can take, as the origin of μ, the point of intersection between h_{12} and D encountered first when marching along D. In subcases* 3.2 *and* 4.2, *we can take, as the origin of μ', the point of intersection between h_{12} and D' encountered first when marching along D'.*

Let $\mathscr{H}$ be the current set of polygonal chains $h_{i,i+1}$ between pairs of points on the boundary of C that are consecutive at this stage.

LEMMA 3.6. *$\mathscr{H}$ can be stored in a dynamic structure of size $O(n)$ such that:*

(i) *The first intersection between an oriented line D (or D') and a polygonal chain $h_{i,i+1}$ can be found in $O(\log n)$ time;*

(ii) *The structure can be updated in $O(\log n)$ time after each new probe.*

Proof. Let us first describe the data structure used to store set $\mathcal{H}$. Each chain $h_{i\,i+1}$ is considered as the concatenation of two subchains $h^{i}_{i\,i+1}$ and $h^{i+1}_{i\,i+1}$. $h^{i}_{i\,i+1}$ (respectively, $h^{i+1}_{i\,i+1}$) consists of the portion of $h_{i\,i+1}$ belonging exclusively to the ray l_i (respectively, l_{i+1}). Let $s_{p_i p_{i+1}}$ be the point of $h_{i\,i+1}$ where rays l_i and l_{i+1} separate. The two subchains $h^{i}_{i\,i+1}$ and $h^{i+1}_{i\,i+1}$ are themselves decomposed into an ordered list of subchains (called subsubchains) as follows. The first subsubchain is the polygonal line between $s_{p_i p_{i+1}}$ and the first local extremum with respect to the x-axis, encountered when marching along the chain from $s_{p_i p_{i+1}}$ towards the measured point. The last subsubchain is the polygonal-line between the last local extremum and the measured point. The other subsubchains join two successive local extremum. Figure 15 illustrates these definitions: h_{12} is decomposed into two subchains $h^1_{12} = s_{p_1 p_2} \to p_1$ and $h^2_{12} = s_{p_1 p_2} \to p_2$. h^1_{12} is decomposed into five subsubchains, $s_{p_1 p_2} \to e^1_1$, $e^1_i \to e^1_{i+1}$ $(i = 1, 2, 3)$, $e^1_4 \to p_1$. h^2_{12} is decomposed into five subsubchains, $s_{p_1 p_2} \to e^2_1$, $e^2_i \to e^2_{i+1}$ $(i = 1, 2, 3)$, $e^2_4 \to p_2$. Each of these subsubchains is stored in a concatenable queue. Since the total number of segments of the polygonal chains $h_{i\,i+1}$ is $O(n)$, the whole data structure can be implemented so as to require $O(n)$ space.

Let us show that this structure allows us to achieve the two goals of the lemma. Let D be an oriented line. Since $h^{i}_{i\,i+1}$ and $h^{i+1}_{i\,i+1}$ are convex, it is clear that D intersects each new subsubchain in at most two points and that these intersections can be computed in $O(\log n)$ time by binary search. Furthermore, we claim that the first intersection between D and $h_{i\,i+1}$, encountered when marching along D, belongs to the first three subsubchains of either $h^{i}_{i\,i+1}$ or $h^{i+1}_{i\,i+1}$. Indeed, let us consider the convex hull $CH(h_{i\,i+1})$ of the chain $h_{i\,i+1}$:

(i) The point $s_{p_i p_{i+1}}$ belongs to $CH(h_{i\,i+1})$ as is easily shown by induction.

(ii) From the convexity of $h_{i\,i+1}$, $CH(h_{i\,i+1})$ is in fact a part of $h_{i\,i+1}$ plus one bridging edge.

(iii) Because D intersects the segment $p_i p_{i+1}$ leaving point p_i on its left side and p_{i+1} on its right side,† D intersects $h_{i\,i+1}$ for the first time in the portion of this chain that belongs to $CH(h_{i\,i+1})$.

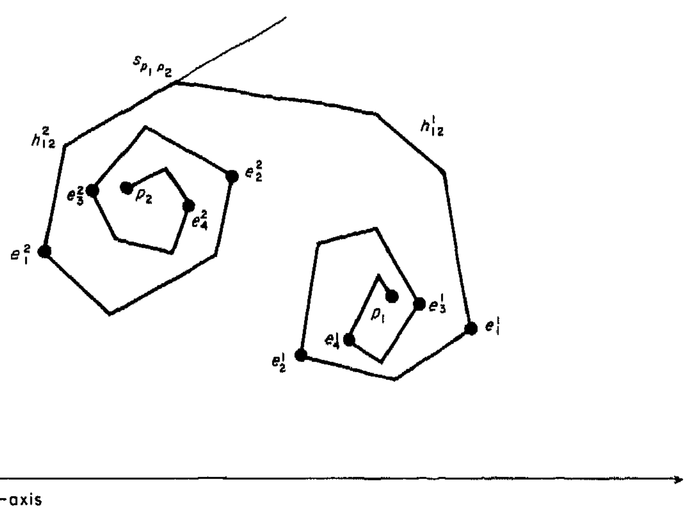

Figure 15. For the notion of subsubchains.

† In the case of the additional probe μ', needed in Case 3.2 (respectively, Case 4.2), the same argument holds, provided that we rename p_i (respectively, p_{i+1}) the corner I, intersection of D_i and D_{i+1}.

(iv) The portion of $h_{i\,i+1}$ that belongs to $CH(h_{i\,i+1})$ can only include edges of the first three subsubchains of $h^i_{i\,i+1}$ and $h^{i+1}_{i\,i+1}$.
This proves the claim. Thus to find the first intersection of D with $h_{i\,i+1}$, it is sufficient to search the three first concatenable queues of $h^i_{i\,i+1}$ and $h^{i+1}_{i\,i+1}$. This proves the first part of the lemma.

Let p be a new contact point, between p_i and p_{i+1}. The data structure has to be updated in order to include the four subchains $h^{p_i}_{p_i p}$, $h^{p}_{p_i p}$ (corresponding to the pair (p_i, p)) and $h^{p}_{p\,p_{i+1}}$, $h^{p_{i+1}}_{p\,p_{i+1}}$ (corresponding to the pair (p, p_{i+1})) and to remove the subchains $h^i_{i\,i+1}$ and $h^{i+1}_{i\,i+1}$. Without loss of generality, assume for instance that the point O where the ray measuring p separates from l_i or l_{i+1}, belongs to $h^i_{i\,i+1}$. To update the data structure, we need to perform the following steps:

(i) The polygonal chain $h^i_{i\,i+1}$ is cut into two parts at point O. This entails splitting the concatenable queue associated with the subsubchain containing O.

(ii) The new chain $h^{p_i}_{p_i p}$ is the part of $h^i_{i\,i+1}$, denoted $h^{i(2)}_{i\,i+1}$, that joins O to p_i.

(iii) The chain $h^{p}_{p_i p}$ consists of only one subsubchain, namely the segment Op. $s_{p_i p}$ is identical to O.

(iv) The chain $h^{p}_{p\,p_{i+1}}$ is formed by concatenating the first part $h^{i(1)}_{i\,i+1}$ of $h^i_{i\,i+1}$ (joining $s_{p_i p_{i+1}}$ to O) and segment Op. $s_{p\,p_{i+1}}$ is identical to $s_{p_i p_{i+1}}$. In this operation, we have to check whether or not O is a local extremum of $h^{p}_{p\,p_{i+1}}$. If O is not a local extremum, Op is simply appended to the last subsubchain of $h^{i(1)}_{i\,i+1}$; if O is a local extremum, Op itself is a subsubchain, namely the last subsubchain in the list of subsubchains of $h^{i(1)}_{i\,i+1}$.

(v) The chain $h^{p_{i+1}}_{p\,p_{i+1}}$ is simply the previous chain $h^{p_{i+1}}_{p_i p_{i+1}}$.
Checking for a local extremum can be done in constant time while splitting or concatenating concatenable queues can be done in $O(\log n)$ time. Thus, updating the data structure, after the probing of a new point p, can be done in $O(\log n)$ time. This proves the second part of the lemma.

We can state now the main result of this section:

THEOREM 3.7. *Each of the at most* $3n-3$ *probes can be determined in* $O(\log n)$ *time yielding an algorithm with overall complexity* $O(n \log n)$ *time and* $O(n)$ *storage for discovering the exact shape of a simple polygon with n non-colinear edges.*

PROOF. In each of the cases 1, 2 or 3 of section 3.2, the probe path μ is defined as soon as its origin O is found. From Lemma 3.6, the origin O of μ and thus μ itself can be computed in $O(\log n)$ time. In subcase 3.1 (respectively, 4.1), the second probe μ' can be taken to be the bisector of U (respectively, S) with its origin at I and thus can be computed in constant time. In subcases 3.2 and 4.2, it follows from Lemma 2.5 that the origin O' of μ' and thus μ' itself can be computed in $O(\log n)$ time.

When a probe with outcome $\varpi = (p, l, n)$ discovers a new edge between two edges e_1 and e_2, the two pairs (p_1, p) and (p, p_2) are new pairs of consecutive points in the order induced by the rays. Thus updating the list of corners takes constant time.

For each pair of consecutive contact points (in the order induced by the rays, the corresponding polygonal line h_{12} is stored as a concatenable queue. Updating the data structure, when the new polygonal lines h_{1p} and h_{p2} are constructed, can be done in $O(\log n)$ time. With Theorem 2.3, we conclude that the overall time complexity of the algorithm is $O(n \log n)$ time.

At each step of the algorithm, the number of ray segments is equal to the number of contact points already probed which is known to be less than $3n-3$. Thus the total number of edges of all the polygonal lines $h_{i\,i+1}$ is at most $6n-6$ and the storage complexity of the algorithm is $O(n)$.

It is to be noticed that Theorem 3.7 holds though we cannot bound by a constant the number of segments of a ray, which may be $O(n)$. This is obtained by expressing our probes in an implicit way, as the concatenation of a prefix plus an extra segment. Specifying all probes individually would require $\Omega(n^2)$ space and time.

3.4. LOWER BOUND ON THE NUMBER OF PROBES

Remember that, in our model, a probe tangent to an edge misses that edge. Thus polygon C is not completely explored as long as we do not have a contact point on each edge and a contact point at each vertex. Indeed otherwise, we might have missed an edge of C (of potentially arbitrary small size). Thus a trivial lower bound on the number of probes needed to discover an n-sided polygon is $2n$. Our algorithm meets this bound for convex polygons. For non-convex polygons, we show, in this section, that whatever the probing strategy may be, $3n-3$ probes may be necessary in the worst-case, which proves that the results of the two previous sections are optimal.

Let us consider a probing strategy S that tries to discover the exact shape of C by means of a sequence of probes. Suppose that $i-1$ probes have already been performed. At this stage, some edges of C have been discovered. The rays associated with the probes induce an ordering of the discovered edges (the same as the one on C). As in section 3.2, the intersection I between the supporting lines D_1 and D_2 of two successive contact points is called a corner and is a potential vertex of C. Either this corner is a vertex of C or some new edge has to be discovered between edge e_1 and edge e_2. Sooner or later, S will have to issue a probe aiming at I in order to decide whether this corner is an actual vertex of C or not. When this probe answers with a point that coincides with I but belongs to only one of the edges e_1 or e_2, no new edge has been discovered and no vertex has been confirmed. Our objective is to construct a polygon C where this adverse situation is encountered $n-3$ times.

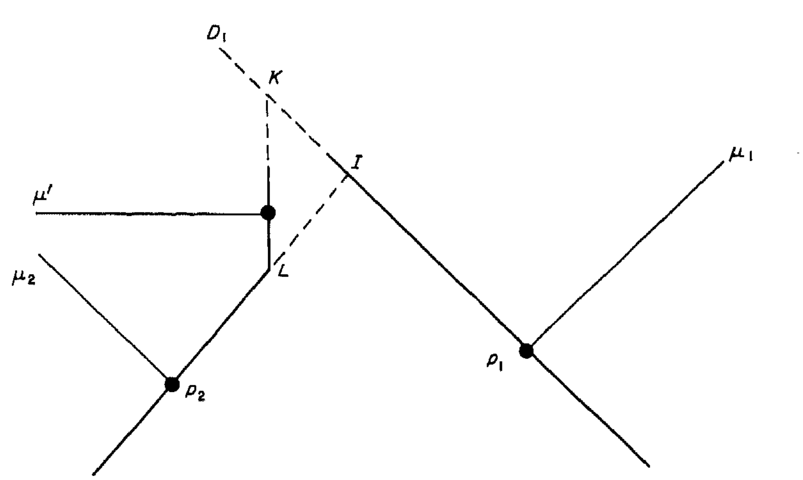

Figure 16. The lower bound.

The construction is done by induction on the number of corners which are actual vertices of C. Let $\hat{C}$ be the current estimate of C i.e., $\hat{C}$ passes through all the contact points, vertices and corners found so far, in the order induced by the rays. At each step, C is chosen so that $\hat{C}$ satisfies the following induction hypothesis:

INDUCTION HYPOTHESIS: *All the corners of $\hat{C}$ are actual vertices of C except one, which is a convex corner (a corner between two successive points p_i and p_{i+1} is convex if the angle (n_i, n_{i+1}) between the oriented normals is in $[0, \pi]$ modulo 2π).*

In the best case, three initial probes p_1, p_2, p_3 belong to three distinct edges. We can always choose C so that the three corners are convex, two of them being actual vertices of C—a situation that fulfills the hypothesis.

Let us suppose that, at a given stage, $\hat{C}$ has $k+1$ vertices and satisfies the induction hypothesis. Let I be the convex corner of $\hat{C}$ that is not a vertex of C. I is the intersection of the supporting lines of D_1 and D_2 of the successive contact points p_1 and p_2. We can always choose C so that the two following conditions are satisfied:

(i) at least two probes are necessary to discover a new edge e (with supporting line D_e) between p_1 and p_2;

(ii) one of the two new corners (D_1, D_e) or (D_e, D_2) is actually a vertex of C while the other is a convex corner.

Let μ be the probe aiming at discovering e. Let us first show that μ has to aim at I. We first notice that the origin of μ cannot belong to the interior of $\hat{C}$, otherwise we would choose C to contain also this origin. Moreover, the inductive hypothesis implies that, from p_2 to p_1, the portions C_{21} and $\widehat{C_{21}}$ of, respectively, C and $\hat{C}$ coincide. Therefore, if μ does not intersect the triangle p_1Ip_2, μ either hits the object on C_{21} or we choose C so that μ misses the object. In both cases, no new edge is discovered and the probe is useless; so we further assume that μ intersects p_1Ip_2. Let p be the first intersection between μ and p_1Ip_2. If p is different from I and belongs to D_i $(i=1$ or $2)$, we take e_i to be long enough to contain p. Thus no new edge has been discovered. So let us suppose that the probe aims at I. In a neighbourhood of I, μ is contained in at least one of the half-planes on the right of the supporting lines D_i $(i=1$ or $2)$. We take the corresponding edge e_i to be long enough to contain I so that the contact point coincides with I but is not a vertex of C. Again, no new edge has been discovered.

Let us now show that we can choose the edge e, which is to be discovered next, in such a way that we can restore the inductive hypothesis at the next stage. Remember that, in the adverse situation we consider, either e_1 or e_2 contains the corner I. Assume that e_1 contains I while e_2 does not (the other case is quite similar). Let us choose a point K on the half line supported by D_1, originating at I and that does not contain p_1 and let us choose a point L on the segment p_2I supported by D_2. Let us consider the closed curve $\widehat{C'}$ formed by the segment p_1K, the segment KL, the segment Lp_2 and the portion C_{21} of the boundary of C. Let μ' be a probe that discovers a new edge e between p_1 and p_2. The origin of μ' cannot belong to the interior of $\widehat{C'}$ and moreover, μ' must necessarily intersect $\widehat{C'}$; otherwise C is chosen so that μ' misses the object. Moreover, μ' must hit $\widehat{C'}$ for the first time at a point p of KL; otherwise, either μ' intersects C_{21}, in which case no new edge is discovered or μ' intersects IK (respectively, Lp_2), in which case we choose the length of edge e_1 (respectively, e_2) so that μ' intersects e_1 (respectively, e_2) and thus does not discover any new edge. Point p will be the contact point of the probe μ'. The edge e containing p is chosen to be supported by line (KL) and to have

L as one of its end points. This implies that L is an actual vertex of C. After this probe, all the vertices of the current estimate $\hat{C}$ are actual vertices of C, except for the point K, which is a convex corner. This achieves the inductive proof.

Let us evaluate the number of probes necessary to discover C. Three initial probes are needed to discover the first three edges of C. Then, each new edge requires two additional probes to be discovered. Lastly, n more probes are needed to probe the vertices of C. Thus we have:

THEOREM 3.8. *Every probe algorithm that determines the shape of a polygon with n edges makes at least* $3n-3$ *probes in the worst-case.*

3.5. ABOUT THE PROBE MODEL

3.5.1. NON-STRAIGHT PROBE PATHS

We have assumed, throughout this paper, that the probe paths are issued along straight lines. This is not necessary, except to ensure the convexity of the curves $h_{i,j}$. This property is crucial to compute a new probe path in $O(\log n)$ time but is not necessary to the probing strategy described in section 3.2. In fact, any probe path can be used provided that it passes through the corner I to be checked, it does not intersect the supporting lines and its probe segment lies inside the corresponding $H_{i,j}$ region. The last requirement can still be satisfied for a simple probe path, using the technique, based on the Jordan sign sequences, that is developed in section 3.2.

3.5.2. FINGER PROBES

It is to be noted that the fact that $3n-3$ probes are sufficient and necessary, in the worst case, to completely specify the shape of a non-convex polygonal object is strongly related to the kind of probe outcomes that is assumed. The probing algorithm developed by Cole & Yap for convex objects assumes a simple finger probe model whose outcome consists only of the co-ordinates of a point on the boundary of the object but contains no information on the direction of the normal at that point. What can be done in that case?

Without additional hypothesis, the problem of finding the exact shape of non-convex polygons with a finite number of finger probes has no solution. Indeed, even if colinear points are found, we cannot guarantee that they belong to the same edge of C; thus an edge can never be confirmed as an edge of C. Nevertheless, we show that, when no information on the normal directions is available, a variant of our method will almost surely output the exact shape of the object, provided that, in addition to the two conditions stated in Section 2, the following third condition is fulfilled:

CONDITION 3. If the intersection point of the supporting lines D_i and D_j of any pair of edges e_i and e_j of C belongs to C, then it belongs to e_i or e_j.

Roughly speaking, we apply to this new probing problem the same strategy as in Section 2: at any stage, the algorithm maintains a list of the so far probed points, sorted according to the order induced by the rays, and issues a new ray aiming to probe between two successive points.

An edge is said to be *discovered* when one or two points of the edge have been probed. As previously noted, an edge can never be confirmed as an edge of C. However, we say that a contact point belongs to an *almost surely* (a.s. for short) *confirmed edge* when it belongs to a triplet of consecutive colinear points. This definition is justified by the fact that, because C has no colinear edge, the situation where three points p_1, p_2, p_3, not belonging to the same edge of C, are colinear is unlikely and not stable, i.e. disappears if we slightly move one of the points along C. More precisely, we have the following lemma:

LEMMA 3.9. *Let $\tilde{C}$ be the submanifold of the 3-dimensional manifold $C \times C \times C$ consisting of all triples (p_1, p_2, p_3) of points belonging to C but not to the same edge of C. Let f be the mapping*

$$(p_1, p_2, p_3) \in \tilde{C} \xrightarrow{f} \begin{vmatrix} x_1 - x_2 & x_3 - x_2 \\ y_1 - y_2 & y_3 - y_2 \end{vmatrix} \in \mathfrak{R}$$

$f^{-1}(0)$ is a 2-dimensional linear submanifold of $\tilde{C}$. In particular, $f^{-1}(0)$ has measure 0.

PROOF. Notice that $f(p_1, p_2, p_3) = 0$ if and only if p_1, p_2 and p_3 are on a same line. It is plain to show that 0 is a regular value of f, i.e. we cannot have simultaneously $f = \delta f = 0$, where δf is the derivative of f. Due to the Preimage theorem (Guillemin & Pollack, 1974, page 21), $f^{-1}(0)$ is a submanifold of $\tilde{C}$, with $\dim f^{-1}(0) = \dim \tilde{C} - \dim \mathfrak{R}$. As $\tilde{C}$ is a manifold of dimension 3, we achieve the proof.

Thus each time three contact points p_1, p_2, p_3 are found to be colinear and such that this situation still holds when a small perturbation of the probe path of one of the contact points is performed, almost surely, p_1, p_2, p_3 belong to the same edge.

It is to be noticed that our probe model cannot detect that a contact point is a vertex. Thus, when a corner is reached by a probe segment, we need additional probes, that will be described below, to confirm the corner as a vertex.

The algorithm starts with three probes that ensure that the three outcoming points belong to at least two distinct edges of the boundary of the polygon: for instance, the first two probes can be straight lines aiming at the origin O from two opposite directions. Then, the algorithm considers the first pair (p_1, p_2) of consecutive contact points that do not belong to the same a.s. confirmed edge and distinguishes three cases:

CASE 1. p_1 and p_2 belong to two a.s. confirmed edges e_1 and e_2 of the boundary of the polygon. In that case, the supporting lines D_1 (respectively, D_2) of e_1 (respectively, e_2) are known but the corner $I = D_1 \cap D_2$ is not yet a confirmed vertex and the situation is similar to that of section 2. More precisely, the corner is probed, as described in section 2, using a probe path included in one of the wedges R or T defined by D_1 and D_2. If the contact point p is distinct from I, then a new edge has been discovered (this case is analogous to Case 2 of section 2). Otherwise, we perform two additional probes inside wedges S and U defined by D_1 and D_2, either as in subcases 3.1 and 4.1 or as in subcases 3.2 and 4.2, depending, as in section 2, on the relative positions of e_1, e_2, p_1 and p_2.

1. $p' = p'' = I$: If e_1 and e_2 are actual edges of C (which happens almost surely), I does not belong to the interior of e_1 nor to the interior of e_2. Moreover, due to Condition 3, I does not belong to another edge of C and so I is the vertex $e_1 \cap e_2$. Thus I is a.s. a vertex of C.

2. At least one of the two points p' or p'' is distinct from I: If e_1 and e_2 are actual edges of C, I is not a vertex of C but a new edge between p_1 and p_2 has been discovered. Thus, in this case, a new edge is a.s. discovered.

CASE 2. One of the points, say p_1 for instance (the other case is quite symmetrical), belongs to an a.s. confirmed edge e_1 while the other, p_2 does not. Let D_1 be the oriented line supporting e_1 and let Π_1^+ (respectively, Π_1^-) be the half-plane to the left (respectively, to the right) of D_1. If p_2 lies in Π_1^+ (respectively, in Π_1^-), the algorithm issues—in a way similar to that of section 3.2—a new probe whose probe path μ is included in $H_{12} \cap \Pi_1^+$ (respectively, in $H_{12} \cap \Pi_1^-$): μ is supported by a line D parallel to D_1, intersecting the segment $p_1 p_2$ and oriented so that p_1 lies on its left side and p_2 on its right side. This new probe yields a contact point p that is between p_1 and p_2 on the boundary of the polygon and does not belong to e_1. If the three points p_1, p and p_2 or the three points p, p_2 and the successor of p_2 are found to be colinear, we a.s. confirm the corresponding segment as an edge of C; otherwise p belongs to an edge that is either not yet discovered or not yet confirmed.

CASE 3. Both points p_1 and p_2 do not yet belong to an a.s. confirmed edge. In this case, a new probe is issued within H_{12} and the outcoming contact point will a.s. confirm an edge if one of the following triple of points (predecessor of p_1, p_1, p) or (p_1, p, p_2) or (p, p_2, successor of p_2) belongs to the same line.

Our procedure will output a simple polygon $\hat{C}$ that almost surely is identical to C.

Let us evaluate the number of probes performed by the above procedure. We need three probes, in the worst-case, to discover a new edge except for the first two that are discovered with one probe each. Thus $2+3(n-2)$ probes are required to discover all the edges. We need two more probes per edge to a.s. confirm the edge, that is $2n$ in total. Lastly, we need three probes to a.s. confirm a vertex. Summing these results, we find that our procedure performs $8n-4$ probes in the worst-case.

The complexity analysis of section 3.3 is still valid here. Indeed, as in section 3.3, each probe is determined by intersecting an oriented line D with a convex curve h_{12} and thus can be computed in $O(\log n)$ time, yielding an algorithm with overall time complexity $O(n \log n)$. We sum up our results in the following theorem:

THEOREM 3.10. *Provided that Conditions* 1, 2 *and* 3 *are fulfilled, the above procedure discovers with at most* $8n-4$ *finger probes a polygon that almost surely is identical to* C. *Each probe can be determined in* $O(\log n)$ *time, yielding an algorithm with overall time complexity* $O(n \log n)$.

We can improve the above procedure when C is convex. Indeed, as soon as three points are colinear, they belong to the same edge and moreover, if a point p is colinear with two points a and b along a line D and is also colinear with two points a' and b' along a line $D' \neq D$, p is a vertex of C. It is plain to adapt the above algorithm so that $4n$ probes are sufficient to discover the exact shape of C.

4. Conclusion

This paper has shown that the information provided by the rays is crucial (though generally neglected) when solving 2-dimensional reconstruction problems. The main property of the rays is that they induce a total order on the measured points when the points belong to a simply connected object. This order has been shown to be computable in optimal time $O(n \log n)$. The algorithm is fully dynamic and allows the insertion or the deletion of a point in $O(\log n)$ time.

From this order a polygonal approximation of the object can be deduced in a straightforward manner. However, if not enough data are available or if the points belong to several connected objects (the data are said to be illegal in this case), this polygonal approximation may not be a simple polygon or may intersect the rays. This can be checked in $O(n \log n)$ time.

Though we have constrained each ray to comprise a bounded number of line segments, the method still works if the rays each consists of a bounded number of simple curves. The complexity results (except for the validity test) hold, provided that a line segment and a curved segment and two curved segments can be intersected in constant time.

The order induced by the rays has also been used to find a strategy for discovering the exact shape of a simple (but not necessarily convex) polygon by means of a minimal number of probes. When each probe outcome consists of a contact point, a ray measuring that point and the normal to the object at the point, we have shown that $3n-3$ probes are necessary and sufficient if the object has n non colinear edges. Each probe can be determined in $O(\log n)$ time yielding an $O(n \log n)$-time $O(n)$-space algorithm. When each probe outcome consists of a contact point and a ray measuring that point but not the normal, the same strategy can still be applied. Under a mild condition, $8n-4$ probes are sufficient to discover a shape that is almost surely the actual shape of the object.

Laurent Baratchart, Richard Cole, Franco Preparata and an anonymous referee are gratefully acknowledged for helpful comments on this paper. Jean-Pierre Merlet is also acknowledged for supplying to us his interactive drawing preparation system J^{P}draw.

References

Alevizos, P. D., Boissonnat, J. D., Yvinec, M. (1987). An optimal $O(n \log n)$ algorithm for contour reconstruction from rays. *Proc. 3rd ACM Symp. on Computational Geometry.* Waterloo, France.

Alevizos, P. D., Boissonnat, J. D., Preparata, F. P. (1988). On the boundary of a union of rays. INRIA Technical Report No. 896.

Bernstein, H. J. (1986). Determining the shape of a convex n-sided polygon by using $2n+k$ tactile probes. *Information Processing Letters* **22**, 255–260.

Boissonnat, J. D. (1984). Geometric structures for three dimensional shape respresentation. *ACM Trans. on Graphics* **3**, 266–286.

Chazelle, B., Guibas, L. J. (1985). Visibility and intersection problems in plane geometry. *First ACM Symposium on Computational Geometry.* Baltimore, USA.

Cole, R., Yap, C. (1987). Shape from probing. *Journal of Algorithms* **8**, 19–38.

Dobkin, D., Edelsbrunner, H., Yap, C. K. (1986). Probing convex polytopes. *Proc. ACM Symp. on Theory of Computing* 424–432.

Guillemin, V., Pollack, A. (1974). *Differential Topology.* Prentice-Hall.

O'Rourke, J. (1986). Uniqueness of Orthogonal Connect-the-dots. Dept. Computer Science, Johns Hopkins University, Baltimore, Maryland.

O'Rourke, J., Booth, H., Washington, R. (1987). Connect-the-dots: a new heuristic. *Comp. Graphics and Image Proc.* **39**, 258–266.

Preparata, F. P., Shamos, M. I. (1985). *Computational Geometry: An Introduction.* Springer-Verlag.

Shamos, M. I., Hoey, D. (1975). Geometric intersection problems. *17th IEEE Symp. on Foundations of Computer Sc.* 208–215.

Skiena, S. S. (1988). Geometric Probing. Ph.D. Thesis, Department of Computer Science, University of Illinois at Urbana Champaign. Tech. Report No. UIUCDCS-R-88-1425.